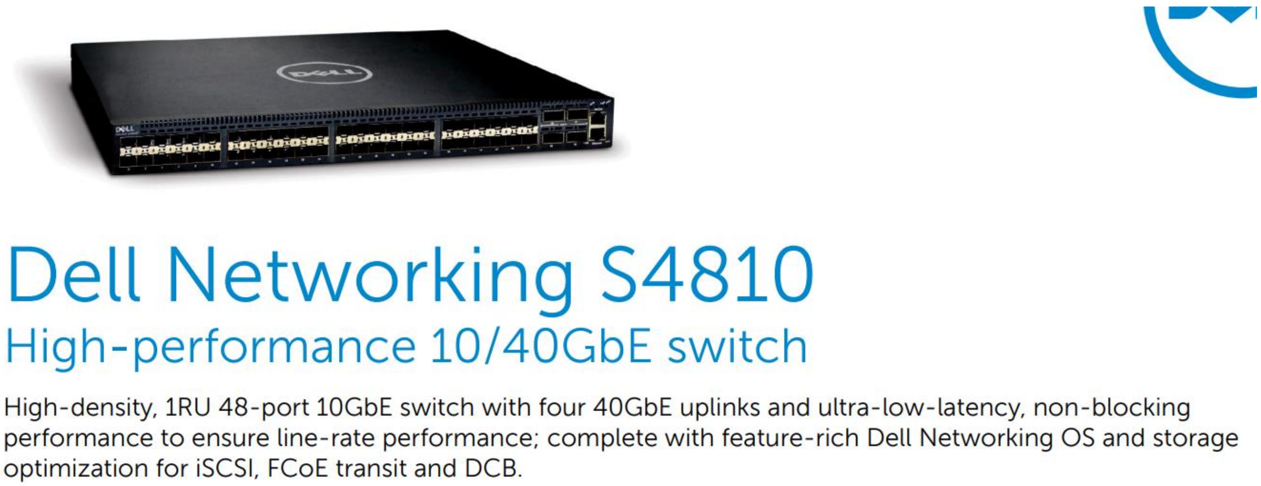
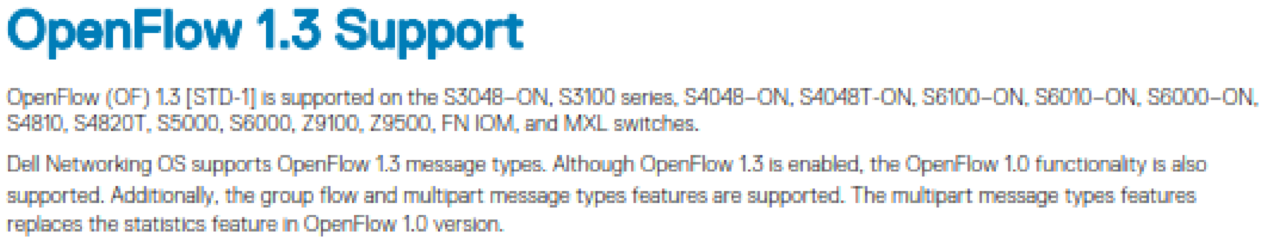
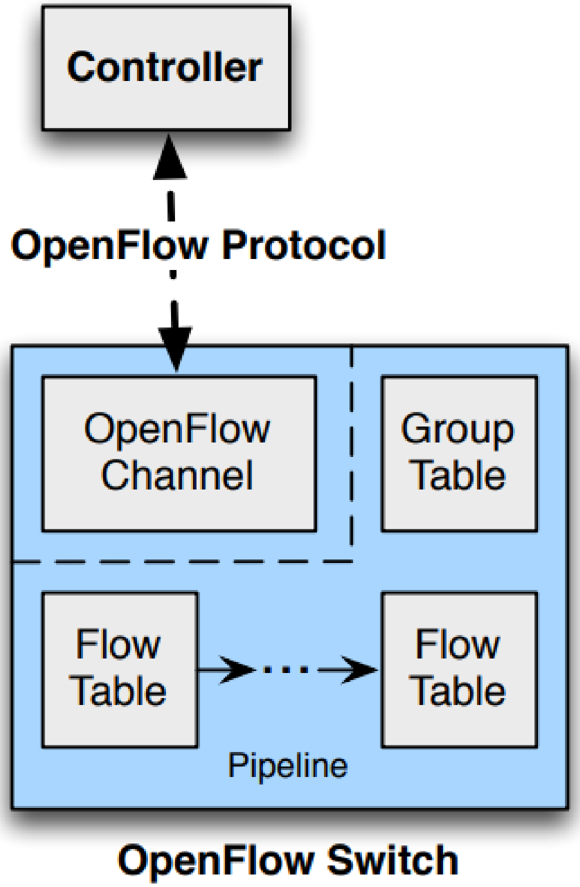
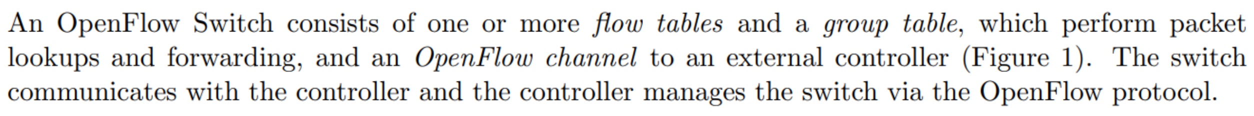
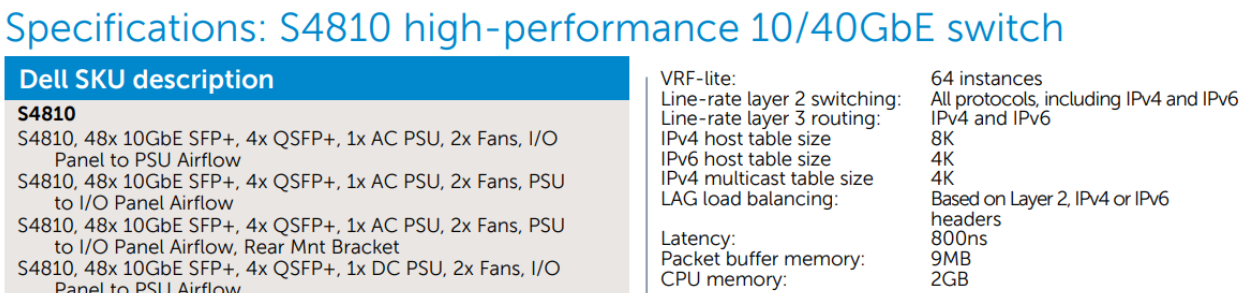
# Exhibit 13

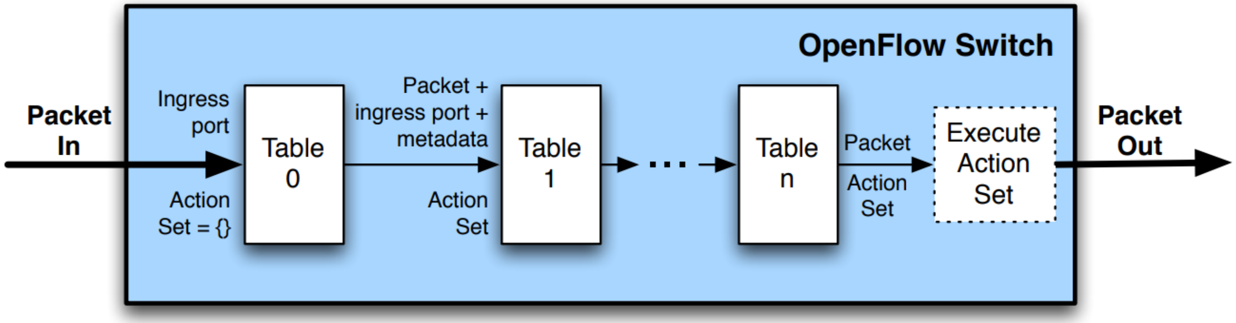
**Illustrative Claim Chart for U.S. Patent No. 9,900,249**

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
| [Preamble] A packet forwarding node, comprising: | The Exemplary Dell Switches are packet forwarding nodes.<br>*See, e.g.*,<br><br>Dell Networking S4810<br>High-performance 10/40GbE switch<br>High-density, 1RU 48-port 10GbE switch with four 40GbE uplinks and ultra-low-latency, non-blocking performance to ensure line-rate performance; complete with feature-rich Dell Networking OS and storage optimization for iSCSI, FCoE transit and DCB.<br>https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/Dell_Networking_S4810_Spec_Sheet.pdf<br>Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches support the OpenFlow specification. |

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
| | <br>OpenFlow 1.3 Support<br>OpenFlow (OF) 1.3 [STD-1] is supported on the S3048–ON, S3100 series, S4048–ON, S4048T-ON, S6100–ON, S6010–ON, S6000–ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches.<br>Dell Networking OS supports OpenFlow 1.3 message types. Although OpenFlow 1.3 is enabled, the OpenFlow 1.0 functionality is also supported. Additionally, the group flow and multipart message types features are supported. The multipart message types features replaces the statistics feature in OpenFlow 1.0 version.<br>Dell OpenFlow Deployment and User Guide 4.0, p. 8. |

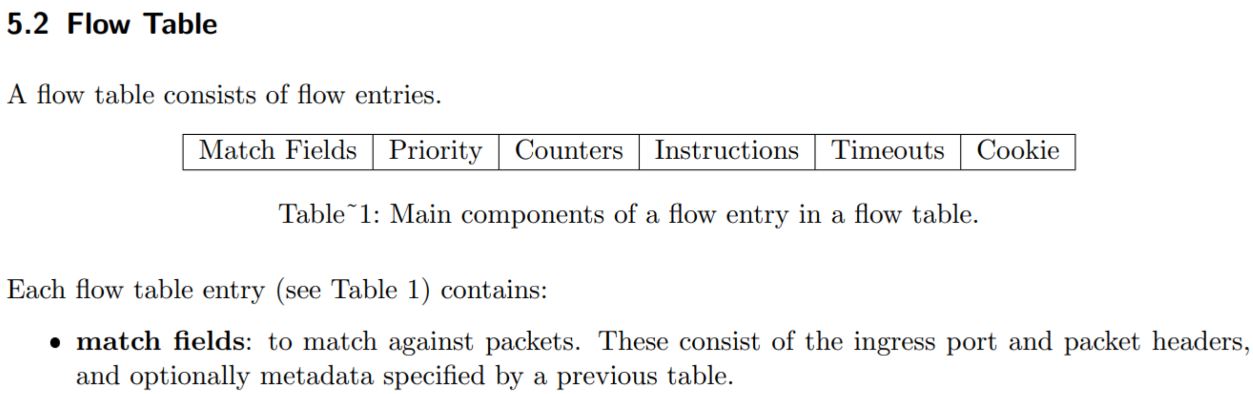
| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
|  |   Controller<br>OpenFlow Protocol<br>OpenFlow Channel<br>Group Table<br>Flow Table<br>Flow Table<br>Pipeline<br>OpenFlow Switch<br>Figure˜1: Main components of an OpenFlow switch.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 8 |

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
| | <br>An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol. |
| [a] a storage unit including a non-transitory computer readable medium configured to store a packet forwarding table used for forwarding a packet based on an identifier which identifies an interface of a packet forwarding node; and | The Exemplary Dell Switches have a storage unit including a non-transitory computer readable medium configured to store a packet forwarding table used for forwarding a packet based on an identifier which identifies an interface of a packet forwarding node.<br><br>*See, e.g.,*<br><br>Specifications: S4810 high-performance 10/40GbE switch<br>Dell SKU description<br>**S4810**<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, I/O Panel to PSU Airflow<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, PSU to I/O Panel Airflow<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, PSU to I/O Panel Airflow, Rear Mnt Bracket<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x DC PSU, 2x Fans, I/O Panel to PSU Airflow<br>VRF-lite: 64 instances<br>Line-rate layer 2 switching: All protocols, including IPv4 and IPv6<br>Line-rate layer 3 routing: IPv4 and IPv6<br>IPv4 host table size 8K<br>IPv6 host table size 4K<br>IPv4 multicast table size 4K<br>LAG load balancing: Based on Layer 2, IPv4 or IPv6 headers<br>Latency: 800ns<br>Packet buffer memory: 9MB<br>CPU memory: 2GB<br>https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/Dell_Networking_S4810_Spec_Sheet.pdf |

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>


(a) Packets are matched against multiple tables in the pipeline

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 13.



5.2 Flow Table

A flow table consists of flow entries.

<table>
<tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr>
</table>

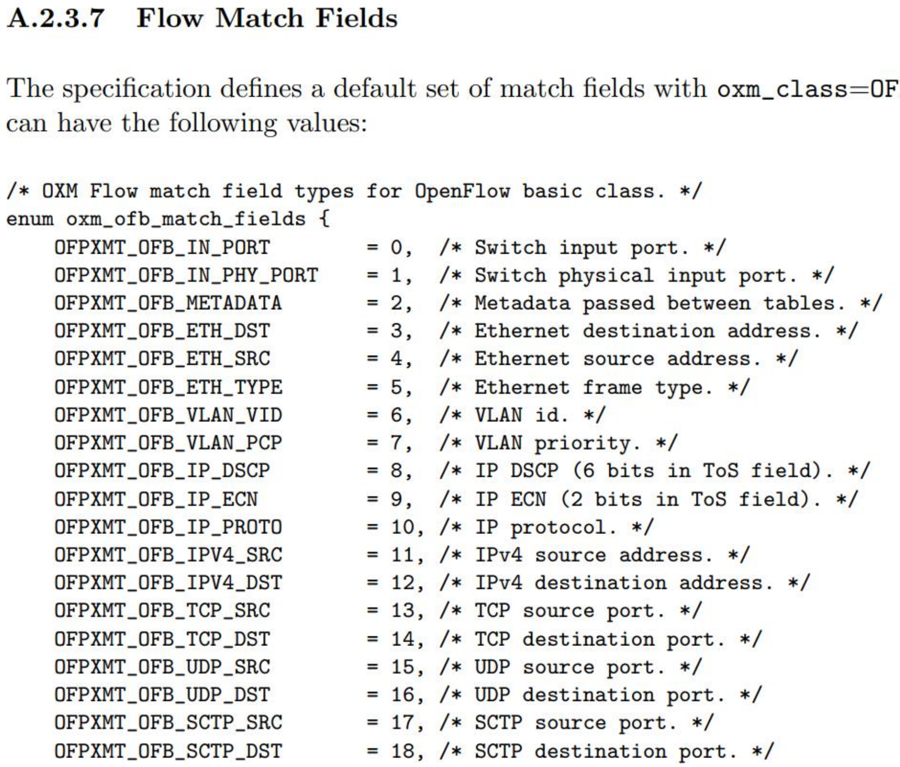
Table˜1: Main components of a flow entry in a flow table.

Each flow table entry (see Table 1) contains:

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.
</td></tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>



A.2.3.7 Flow Match Fields

The specification defines a default set of match fields with oxm_class=OFPXMC_OPENFLOW_BASIC which can have the following values:

```
/* OXM Flow match field types for OpenFlow basic class. */
enum oxm_ofb_match_fields {
    OFPXMT_OFB_IN_PORT        = 0,  /* Switch input port. */
    OFPXMT_OFB_IN_PHY_PORT    = 1,  /* Switch physical input port. */
    OFPXMT_OFB_METADATA       = 2,  /* Metadata passed between tables. */
    OFPXMT_OFB_ETH_DST        = 3,  /* Ethernet destination address. */
    OFPXMT_OFB_ETH_SRC        = 4,  /* Ethernet source address. */
    OFPXMT_OFB_ETH_TYPE       = 5,  /* Ethernet frame type. */
    OFPXMT_OFB_VLAN_VID       = 6,  /* VLAN id. */
    OFPXMT_OFB_VLAN_PCP       = 7,  /* VLAN priority. */
    OFPXMT_OFB_IP_DSCP        = 8,  /* IP DSCP (6 bits in ToS field). */
    OFPXMT_OFB_IP_ECN         = 9,  /* IP ECN (2 bits in ToS field). */
    OFPXMT_OFB_IP_PROTO       = 10, /* IP protocol. */
    OFPXMT_OFB_IPV4_SRC       = 11, /* IPv4 source address. */
    OFPXMT_OFB_IPV4_DST       = 12, /* IPv4 destination address. */
    OFPXMT_OFB_TCP_SRC        = 13, /* TCP source port. */
    OFPXMT_OFB_TCP_DST        = 14, /* TCP destination port. */
    OFPXMT_OFB_UDP_SRC        = 15, /* UDP source port. */
    OFPXMT_OFB_UDP_DST        = 16, /* UDP destination port. */
    OFPXMT_OFB_SCTP_SRC       = 17, /* SCTP source port. */
    OFPXMT_OFB_SCTP_DST       = 18, /* SCTP destination port. */
```

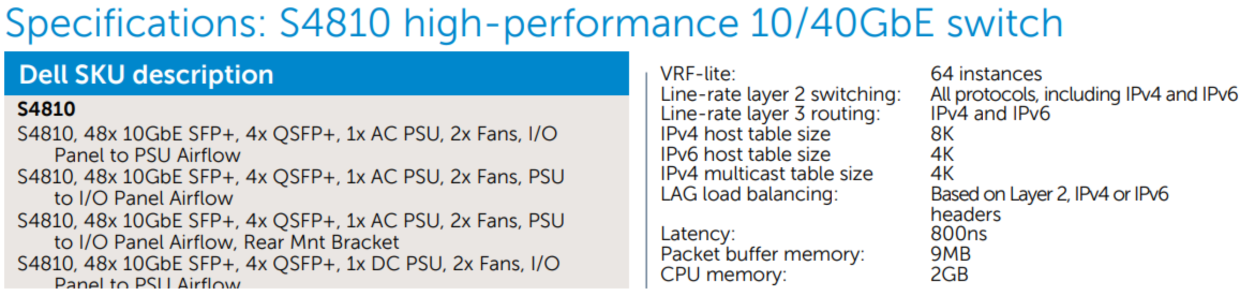
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 49.

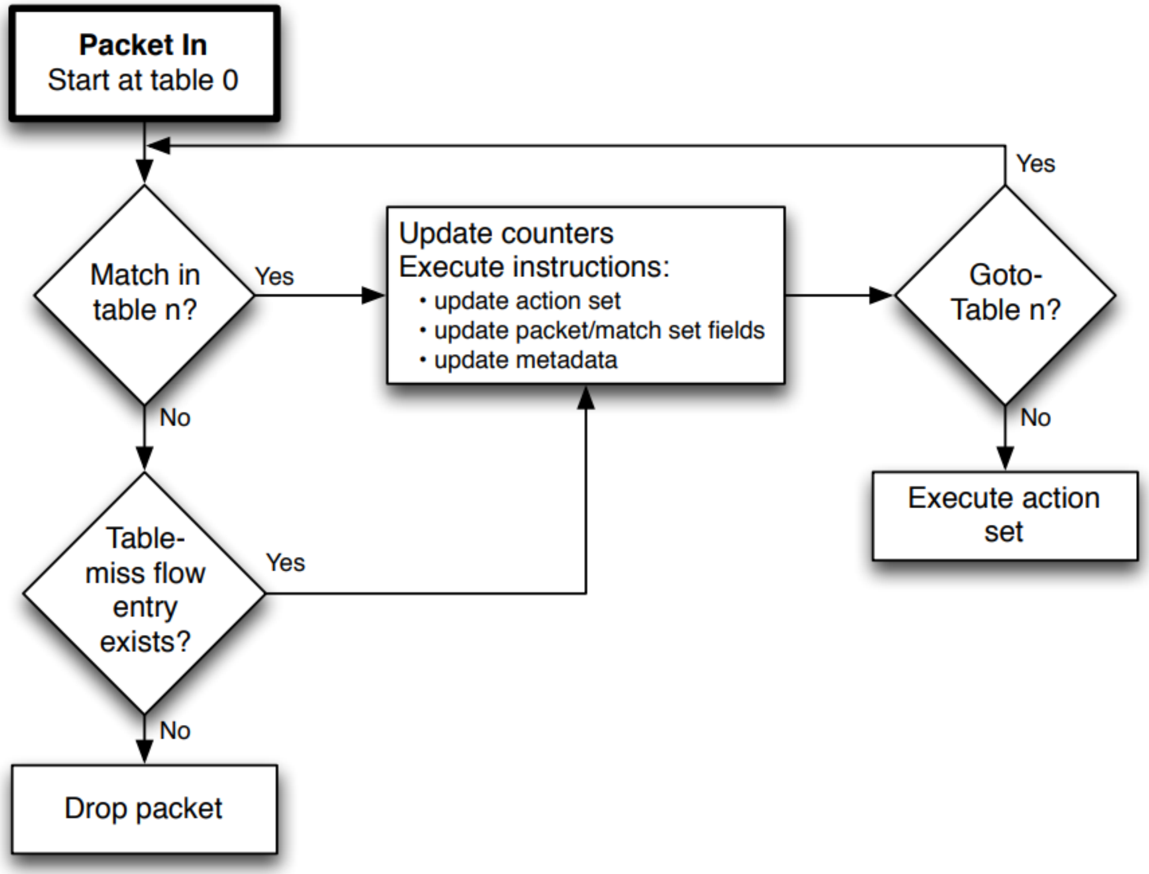
</td></tr>
<tr><td>[b] at least one processor configured to execute program instructions to provide a forwarding unit configured to:</td><td>

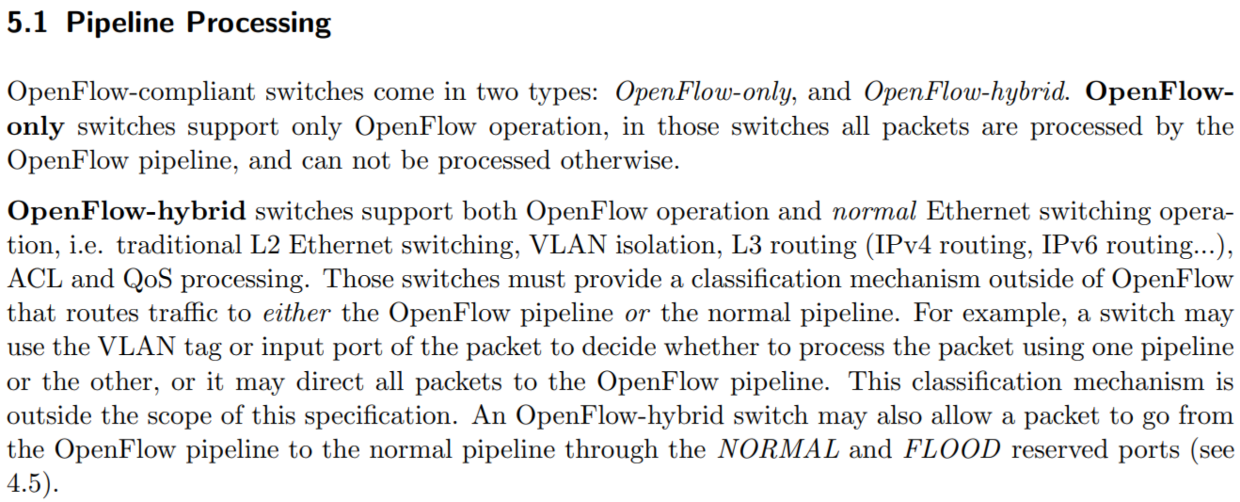
The Exemplary Dell Switches have at least one processor configured to execute program instructions.

*See, e.g.,*

</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
| | <br>Specifications: S4810 high-performance 10/40GbE switch<br>Dell SKU description<br>**S4810**<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, I/O Panel to PSU Airflow<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, PSU to I/O Panel Airflow<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x AC PSU, 2x Fans, PSU to I/O Panel Airflow, Rear Mnt Bracket<br>S4810, 48x 10GbE SFP+, 4x QSFP+, 1x DC PSU, 2x Fans, I/O Panel to PSU Airflow<br>VRF-lite: 64 instances<br>Line-rate layer 2 switching: All protocols, including IPv4 and IPv6<br>Line-rate layer 3 routing: IPv4 and IPv6<br>IPv4 host table size 8K<br>IPv6 host table size 4K<br>IPv4 multicast table size 4K<br>LAG load balancing: Based on Layer 2, IPv4 or IPv6 headers<br>Latency: 800ns<br>Packet buffer memory: 9MB<br>CPU memory: 2GB<br>https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/Dell_Networking_S4810_Spec_Sheet.pdf |
| [c] receive an incoming packet including a plurality of identifiers, | The Exemplary Dell Switches have a forwarding unit configured to receive an incoming packet including a plurality of the identifiers.<br>*See, e.g.*, |

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
|  |  Packet In<br>Start at table 0<br>Match in table n?<br>Yes<br>Update counters<br>Execute instructions:<br>• update action set<br>• update packet/match set fields<br>• update metadata<br>Goto-Table n?<br>Yes<br>No<br>No<br>Execute action set<br>Table-miss flow entry exists?<br>Yes<br>No<br>Drop packet<br>Figure˜3: Flowchart detailing packet flow through an OpenFlow switch.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 15. |

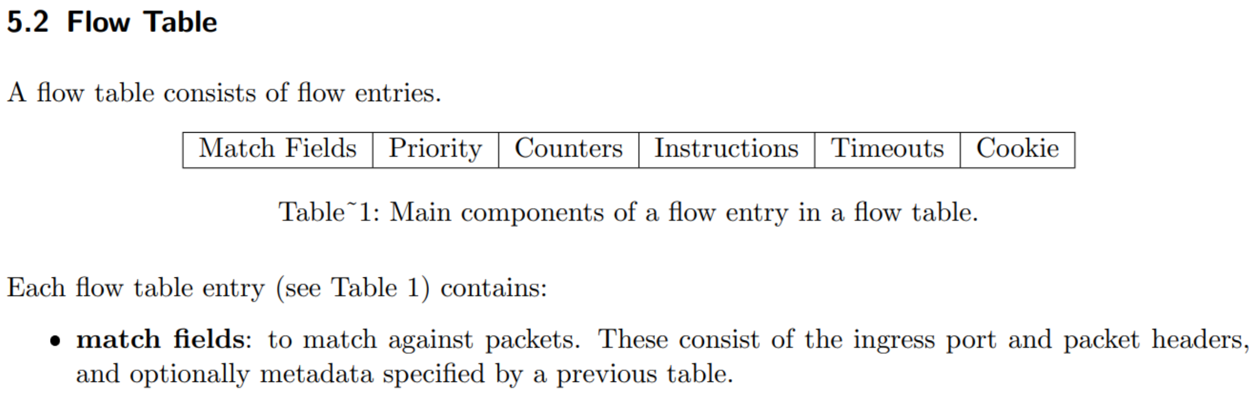
<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>


<b>5.1 Pipeline Processing</b>

OpenFlow-compliant switches come in two types: <i>OpenFlow-only</i>, and <i>OpenFlow-hybrid</i>. <b>OpenFlow-only</b> switches support only OpenFlow operation, in those switches all packets are processed by the OpenFlow pipeline, and can not be processed otherwise.

<b>OpenFlow-hybrid</b> switches support both OpenFlow operation and <i>normal</i> Ethernet switching operation, i.e. traditional L2 Ethernet switching, VLAN isolation, L3 routing (IPv4 routing, IPv6 routing...), ACL and QoS processing. Those switches must provide a classification mechanism outside of OpenFlow that routes traffic to <i>either</i> the OpenFlow pipeline <i>or</i> the normal pipeline. For example, a switch may use the VLAN tag or input port of the packet to decide whether to process the packet using one pipeline or the other, or it may direct all packets to the OpenFlow pipeline. This classification mechanism is outside the scope of this specification. An OpenFlow-hybrid switch may also allow a packet to go from the OpenFlow pipeline to the normal pipeline through the <i>NORMAL</i> and <i>FLOOD</i> reserved ports (see 4.5).

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 13.



<b>5.2 Flow Table</b>

A flow table consists of flow entries.

<table><tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr></table>

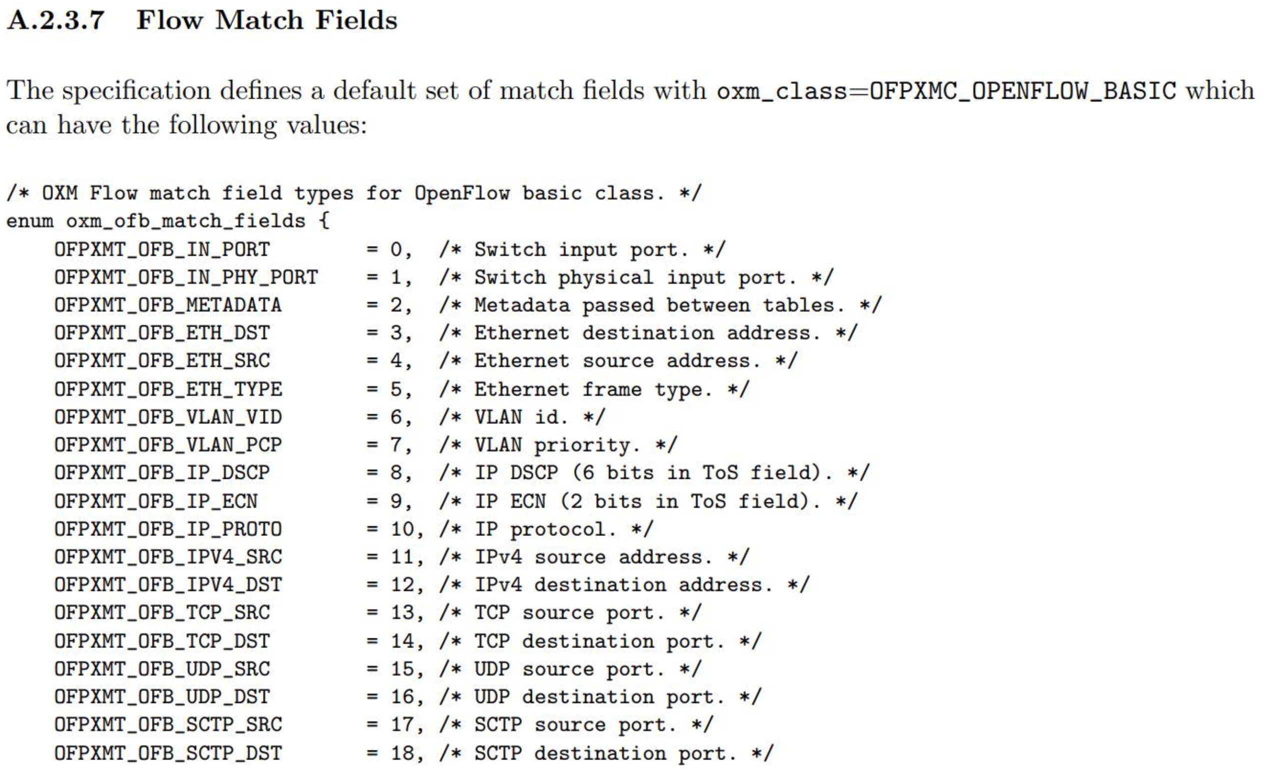
Table˜1: Main components of a flow entry in a flow table.

Each flow table entry (see Table 1) contains:

• <b>match fields</b>: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.
</td></tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>



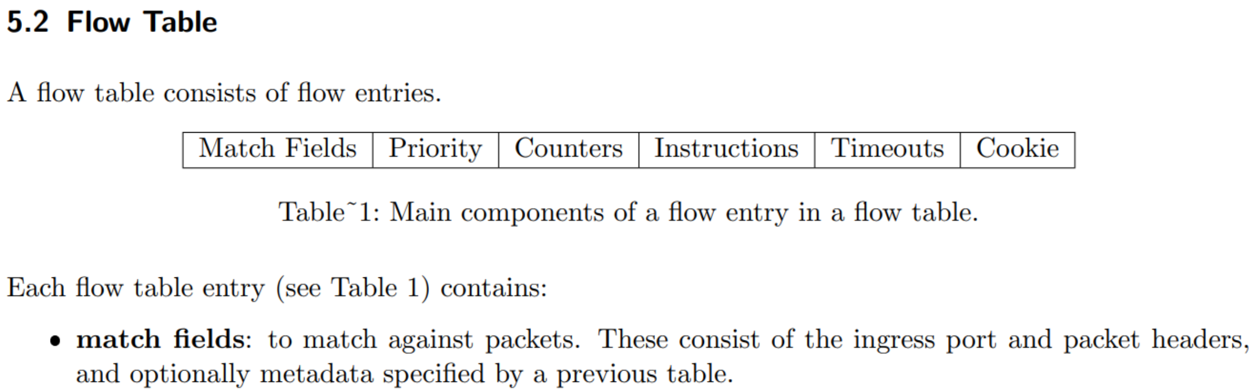
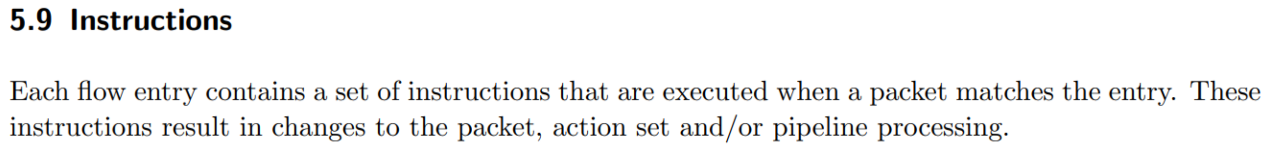
A.2.3.7 Flow Match Fields

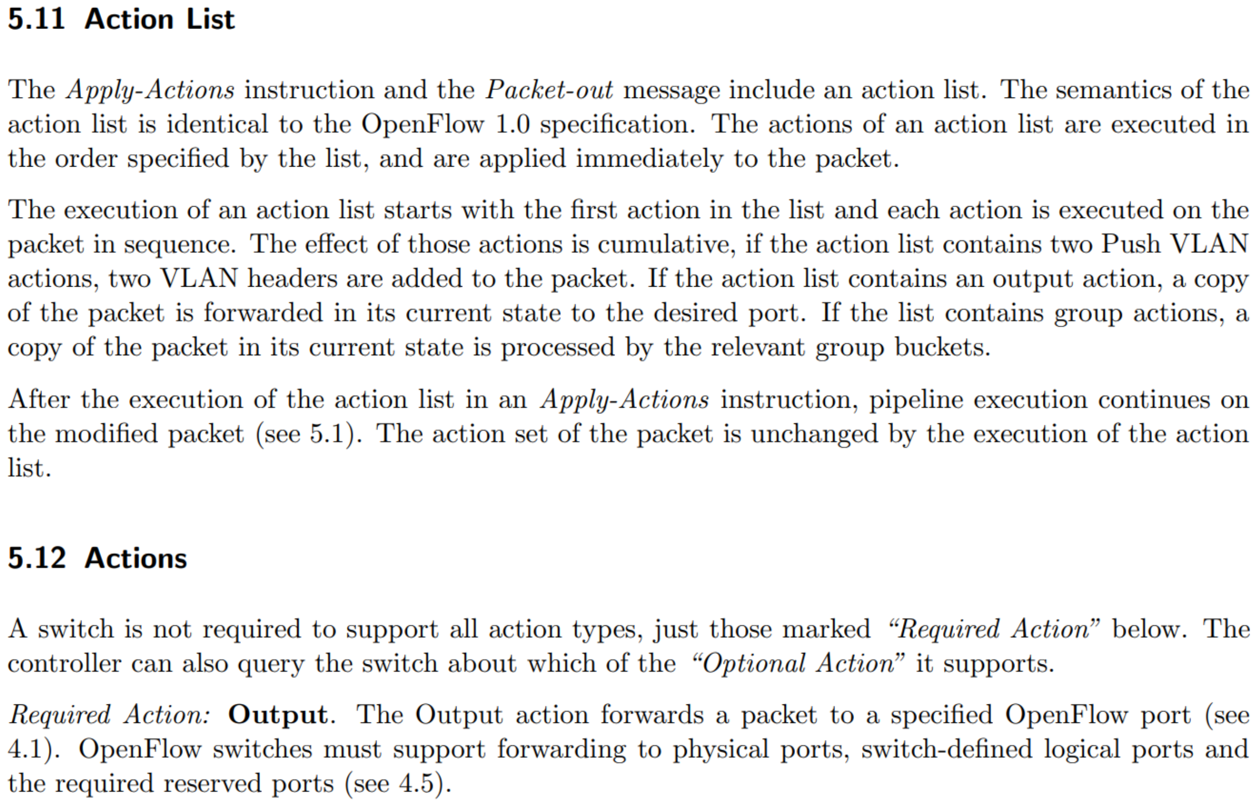
The specification defines a default set of match fields with oxm_class=OFPXMC_OPENFLOW_BASIC which can have the following values:

```
/* OXM Flow match field types for OpenFlow basic class. */
enum oxm_ofb_match_fields {
    OFPXMT_OFB_IN_PORT        = 0,  /* Switch input port. */
    OFPXMT_OFB_IN_PHY_PORT    = 1,  /* Switch physical input port. */
    OFPXMT_OFB_METADATA       = 2,  /* Metadata passed between tables. */
    OFPXMT_OFB_ETH_DST        = 3,  /* Ethernet destination address. */
    OFPXMT_OFB_ETH_SRC        = 4,  /* Ethernet source address. */
    OFPXMT_OFB_ETH_TYPE       = 5,  /* Ethernet frame type. */
    OFPXMT_OFB_VLAN_VID       = 6,  /* VLAN id. */
    OFPXMT_OFB_VLAN_PCP       = 7,  /* VLAN priority. */
    OFPXMT_OFB_IP_DSCP        = 8,  /* IP DSCP (6 bits in ToS field). */
    OFPXMT_OFB_IP_ECN         = 9,  /* IP ECN (2 bits in ToS field). */
    OFPXMT_OFB_IP_PROTO       = 10, /* IP protocol. */
    OFPXMT_OFB_IPV4_SRC       = 11, /* IPv4 source address. */
    OFPXMT_OFB_IPV4_DST       = 12, /* IPv4 destination address. */
    OFPXMT_OFB_TCP_SRC        = 13, /* TCP source port. */
    OFPXMT_OFB_TCP_DST        = 14, /* TCP destination port. */
    OFPXMT_OFB_UDP_SRC        = 15, /* UDP source port. */
    OFPXMT_OFB_UDP_DST        = 16, /* UDP destination port. */
    OFPXMT_OFB_SCTP_SRC       = 17, /* SCTP source port. */
    OFPXMT_OFB_SCTP_DST       = 18, /* SCTP destination port. */
```

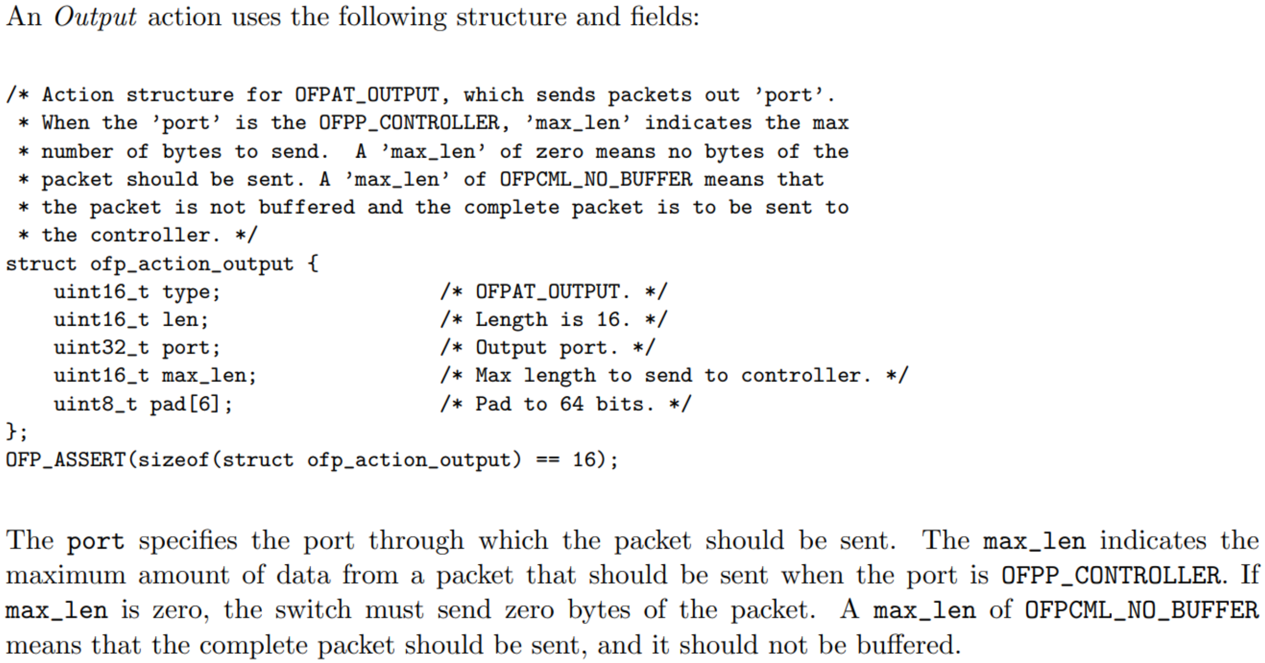
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 49.

</td></tr>
<tr><td>[d] determine a direction to forward the incoming packet,</td><td>The Exemplary Dell Switches have a forwarding unit configured to determine a direction to forward the incoming packet.<br><br><i>See, e.g.</i>,</td></tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td><br>5.2 Flow Table<br><br>A flow table consists of flow entries.<br><table><tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr></table><br>Table˜1: Main components of a flow entry in a flow table.<br><br>Each flow table entry (see Table 1) contains:<br>• match fields: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.<br><br><br>5.9 Instructions<br><br>Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 19.</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
|  | <br>**5.11 Action List**<br>The *Apply-Actions* instruction and the *Packet-out* message include an action list. The semantics of the action list is identical to the OpenFlow 1.0 specification. The actions of an action list are executed in the order specified by the list, and are applied immediately to the packet.<br>The execution of an action list starts with the first action in the list and each action is executed on the packet in sequence. The effect of those actions is cumulative, if the action list contains two Push VLAN actions, two VLAN headers are added to the packet. If the action list contains an output action, a copy of the packet is forwarded in its current state to the desired port. If the list contains group actions, a copy of the packet in its current state is processed by the relevant group buckets.<br>After the execution of the action list in an *Apply-Actions* instruction, pipeline execution continues on the modified packet (see 5.1). The action set of the packet is unchanged by the execution of the action list.<br>**5.12 Actions**<br>A switch is not required to support all action types, just those marked *"Required Action"* below. The controller can also query the switch about which of the *"Optional Action"* it supports.<br>*Required Action:* **Output**. The Output action forwards a packet to a specified OpenFlow port (see 4.1). OpenFlow switches must support forwarding to physical ports, switch-defined logical ports and the required reserved ports (see 4.5).<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 22. |

<table>
<tr>
<th>Claim 6</th>
<th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th>
</tr>
<tr>
<td></td>
<td>



An *Output* action uses the following structure and fields:

```
/* Action structure for OFPAT_OUTPUT, which sends packets out 'port'.
 * When the 'port' is the OFPP_CONTROLLER, 'max_len' indicates the max
 * number of bytes to send.  A 'max_len' of zero means no bytes of the
 * packet should be sent. A 'max_len' of OFPCML_NO_BUFFER means that
 * the packet is not buffered and the complete packet is to be sent to
 * the controller. */
struct ofp_action_output {
    uint16_t type;                  /* OFPAT_OUTPUT. */
    uint16_t len;                   /* Length is 16. */
    uint32_t port;                  /* Output port. */
    uint16_t max_len;               /* Max length to send to controller. */
    uint8_t pad[6];                 /* Pad to 64 bits. */
};
OFP_ASSERT(sizeof(struct ofp_action_output) == 16);
```

The **port** specifies the port through which the packet should be sent. The **max_len** indicates the maximum amount of data from a packet that should be sent when the port is OFPP_CONTROLLER. If **max_len** is zero, the switch must send zero bytes of the packet. A **max_len** of OFPCML_NO_BUFFER means that the complete packet should be sent, and it should not be buffered.

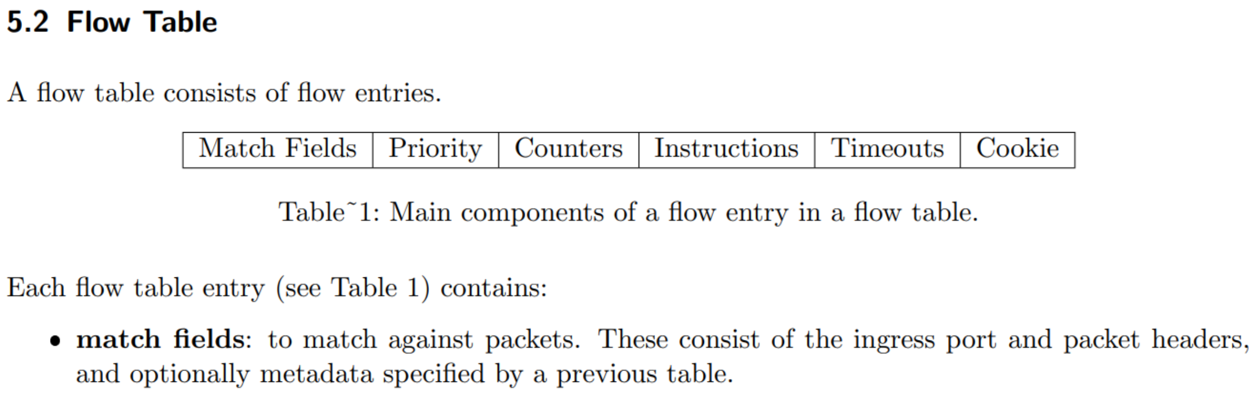
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 57.

</td>
</tr>
<tr>
<td>[e] determine whether to forward to an external network,</td>
<td>

The Exemplary Dell Switches have a forwarding unit configured to determine whether to forward to an external network.

*See, e.g.,*

</td>
</tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>


5.2 Flow Table

A flow table consists of flow entries.

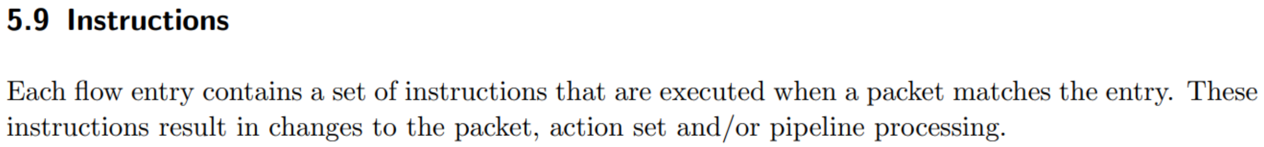
<table><tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr></table>

Table˜1: Main components of a flow entry in a flow table.

Each flow table entry (see Table 1) contains:

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.

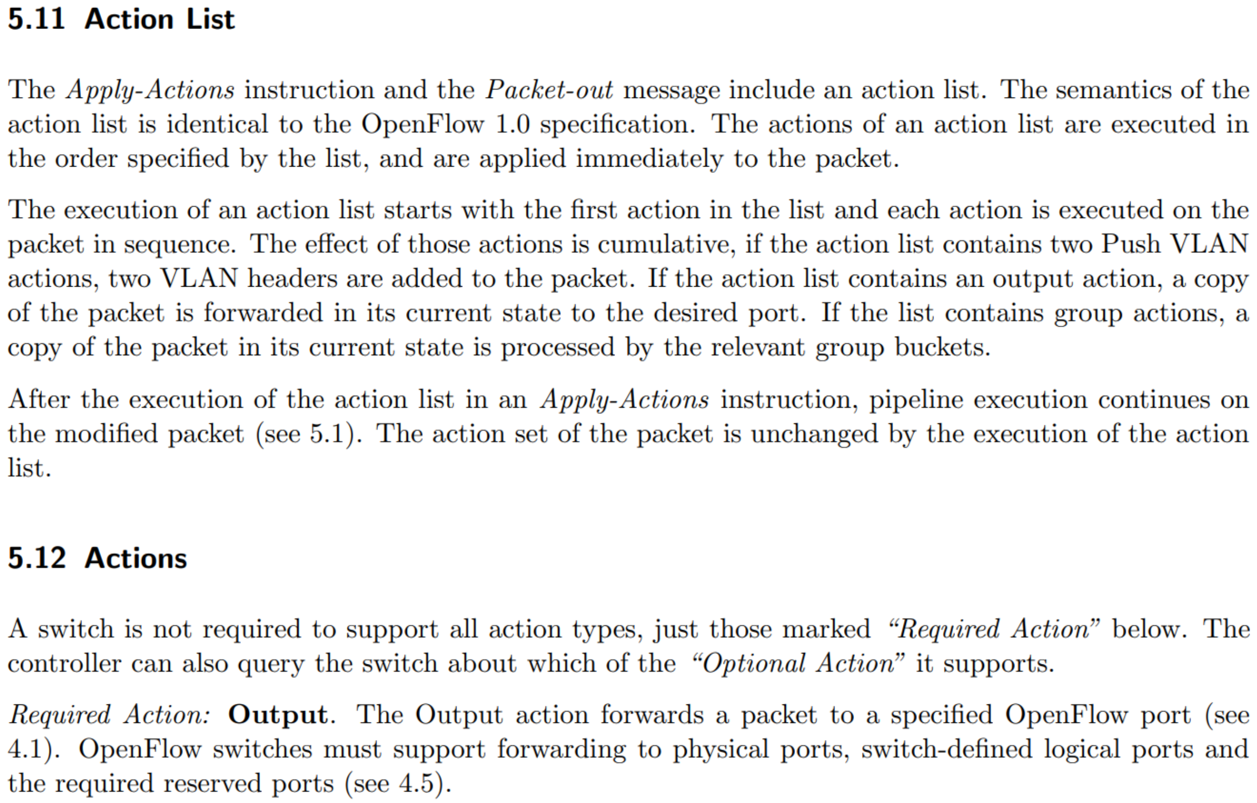
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.



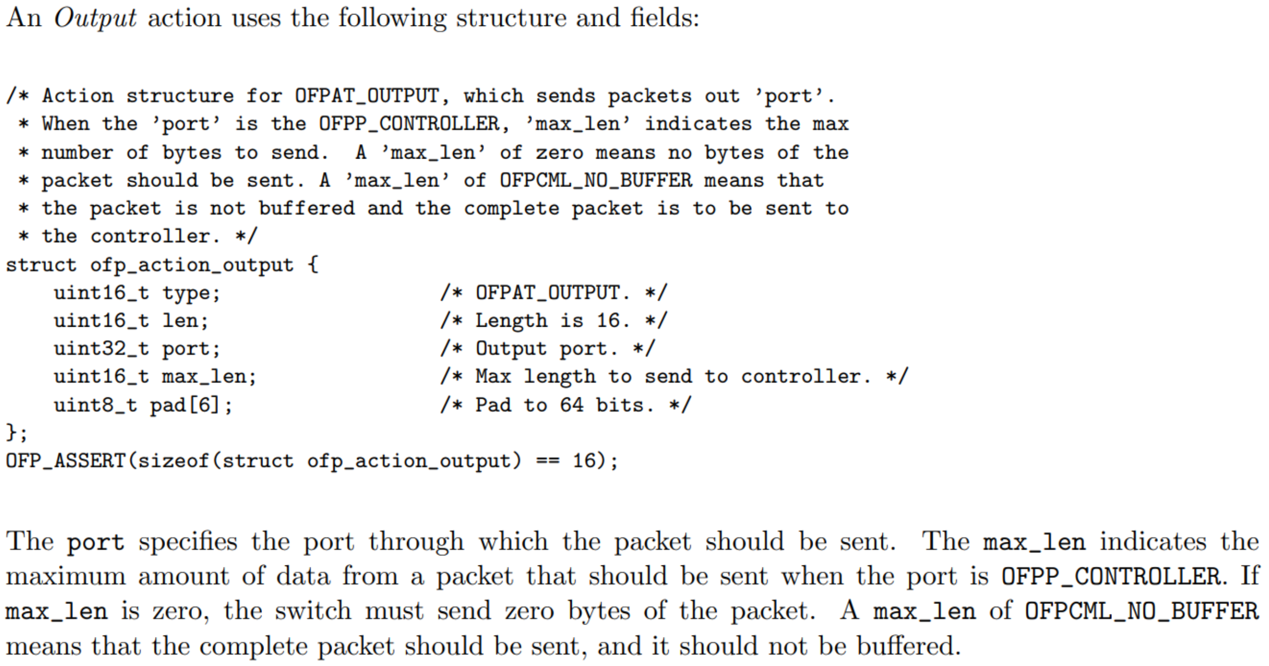
5.9 Instructions

Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 19.
</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
| | <br>**5.11 Action List**<br>The *Apply-Actions* instruction and the *Packet-out* message include an action list. The semantics of the action list is identical to the OpenFlow 1.0 specification. The actions of an action list are executed in the order specified by the list, and are applied immediately to the packet.<br>The execution of an action list starts with the first action in the list and each action is executed on the packet in sequence. The effect of those actions is cumulative, if the action list contains two Push VLAN actions, two VLAN headers are added to the packet. If the action list contains an output action, a copy of the packet is forwarded in its current state to the desired port. If the list contains group actions, a copy of the packet in its current state is processed by the relevant group buckets.<br>After the execution of the action list in an *Apply-Actions* instruction, pipeline execution continues on the modified packet (see 5.1). The action set of the packet is unchanged by the execution of the action list.<br>**5.12 Actions**<br>A switch is not required to support all action types, just those marked *"Required Action"* below. The controller can also query the switch about which of the *"Optional Action"* it supports.<br>*Required Action:* **Output**. The Output action forwards a packet to a specified OpenFlow port (see 4.1). OpenFlow switches must support forwarding to physical ports, switch-defined logical ports and the required reserved ports (see 4.5).<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 22. |

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>



An *Output* action uses the following structure and fields:

```
/* Action structure for OFPAT_OUTPUT, which sends packets out 'port'.
 * When the 'port' is the OFPP_CONTROLLER, 'max_len' indicates the max
 * number of bytes to send.  A 'max_len' of zero means no bytes of the
 * packet should be sent. A 'max_len' of OFPCML_NO_BUFFER means that
 * the packet is not buffered and the complete packet is to be sent to
 * the controller. */
struct ofp_action_output {
    uint16_t type;                  /* OFPAT_OUTPUT. */
    uint16_t len;                   /* Length is 16. */
    uint32_t port;                  /* Output port. */
    uint16_t max_len;               /* Max length to send to controller. */
    uint8_t pad[6];                 /* Pad to 64 bits. */
};
OFP_ASSERT(sizeof(struct ofp_action_output) == 16);
```

The **port** specifies the port through which the packet should be sent. The **max_len** indicates the maximum amount of data from a packet that should be sent when the port is OFPP_CONTROLLER. If **max_len** is zero, the switch must send zero bytes of the packet. A **max_len** of OFPCML_NO_BUFFER means that the complete packet should be sent, and it should not be buffered.

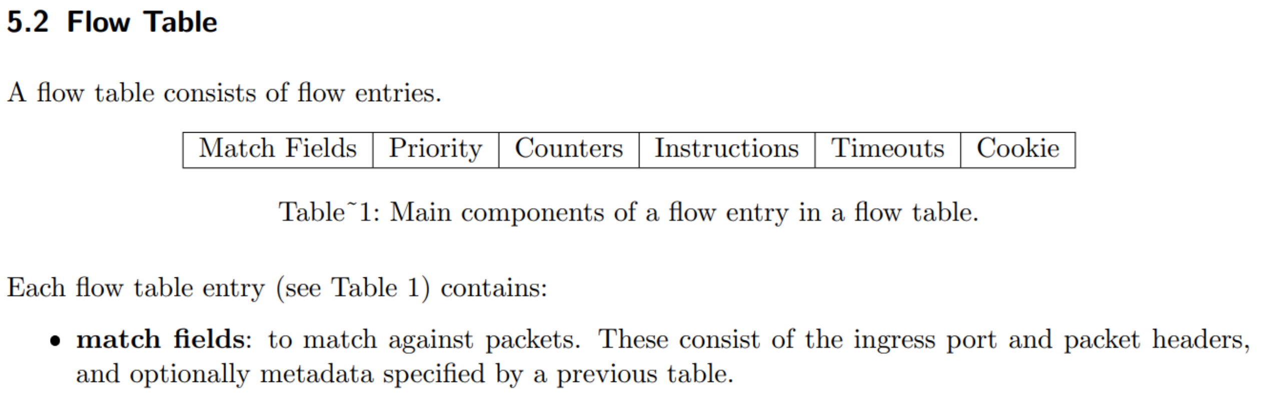
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 57.

</td></tr>
<tr><td>[f] add to path or link information when the incoming packet is not forwarded to the external network,</td><td>

The Exemplary Dell Switches have a forwarding unit configured to add to path or link information when the incoming packet is not forwarded to the external network.

*See, e.g.*,

</td></tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>


### 5.2 Flow Table

A flow table consists of flow entries.

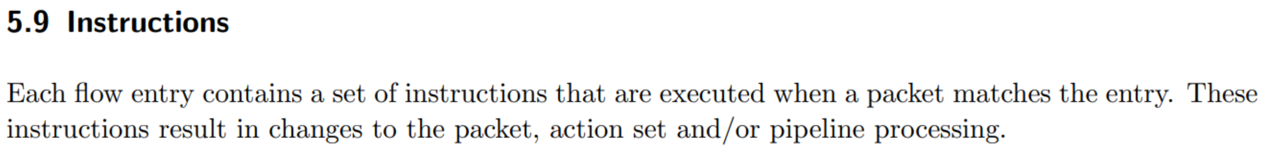
<table>
<tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr>
</table>

Table˜1: Main components of a flow entry in a flow table.

Each flow table entry (see Table 1) contains:

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.

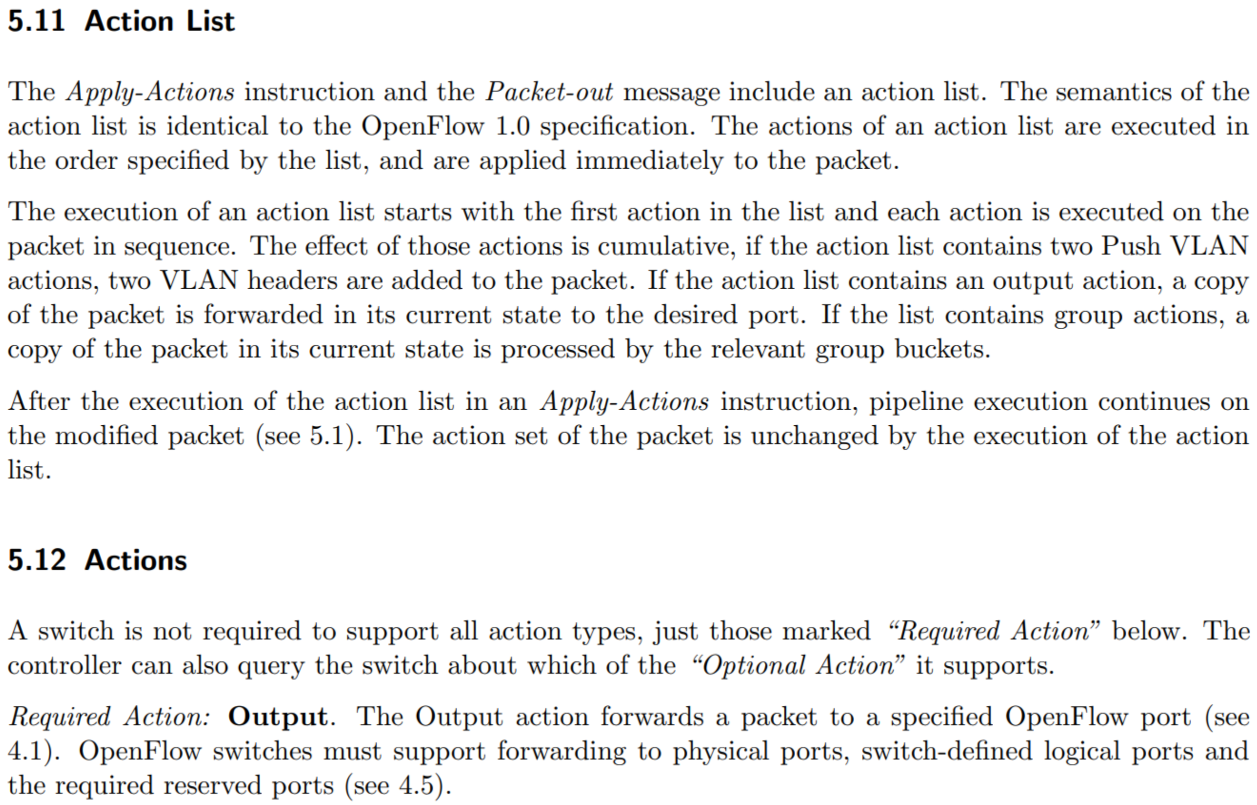
https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.



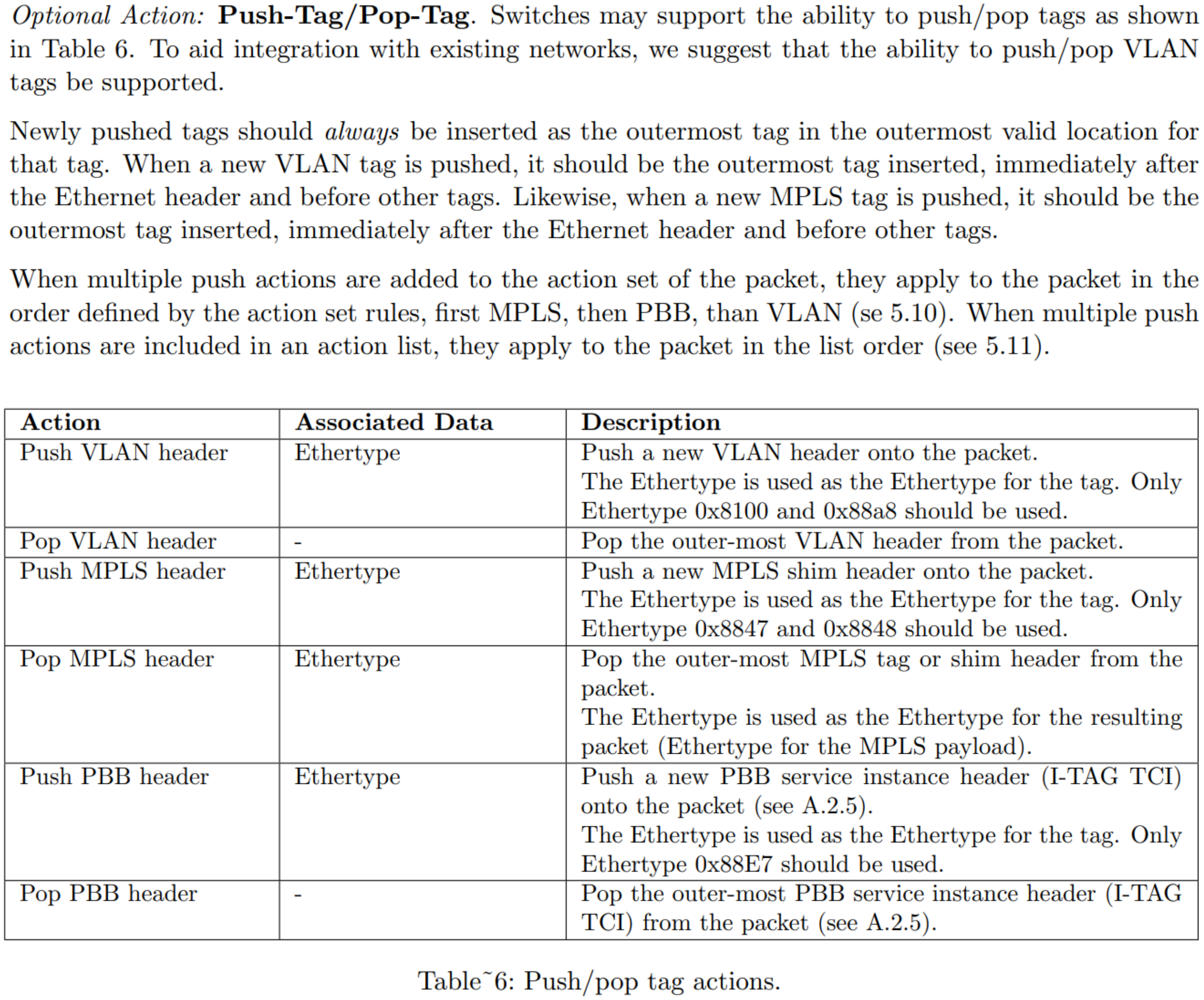
### 5.9 Instructions

Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 19.
</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
|  | <br>**5.11 Action List**<br>The *Apply-Actions* instruction and the *Packet-out* message include an action list. The semantics of the action list is identical to the OpenFlow 1.0 specification. The actions of an action list are executed in the order specified by the list, and are applied immediately to the packet.<br>The execution of an action list starts with the first action in the list and each action is executed on the packet in sequence. The effect of those actions is cumulative, if the action list contains two Push VLAN actions, two VLAN headers are added to the packet. If the action list contains an output action, a copy of the packet is forwarded in its current state to the desired port. If the list contains group actions, a copy of the packet in its current state is processed by the relevant group buckets.<br>After the execution of the action list in an *Apply-Actions* instruction, pipeline execution continues on the modified packet (see 5.1). The action set of the packet is unchanged by the execution of the action list.<br>**5.12 Actions**<br>A switch is not required to support all action types, just those marked *"Required Action"* below. The controller can also query the switch about which of the *"Optional Action"* it supports.<br>*Required Action:* **Output**. The Output action forwards a packet to a specified OpenFlow port (see 4.1). OpenFlow switches must support forwarding to physical ports, switch-defined logical ports and the required reserved ports (see 4.5).<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 22. |

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|



*Optional Action:* **Push-Tag/Pop-Tag**. Switches may support the ability to push/pop tags as shown in Table 6. To aid integration with existing networks, we suggest that the ability to push/pop VLAN tags be supported.

Newly pushed tags should *always* be inserted as the outermost tag in the outermost valid location for that tag. When a new VLAN tag is pushed, it should be the outermost tag inserted, immediately after the Ethernet header and before other tags. Likewise, when a new MPLS tag is pushed, it should be the outermost tag inserted, immediately after the Ethernet header and before other tags.

When multiple push actions are added to the action set of the packet, they apply to the packet in the order defined by the action set rules, first MPLS, then PBB, than VLAN (se 5.10). When multiple push actions are included in an action list, they apply to the packet in the list order (see 5.11).

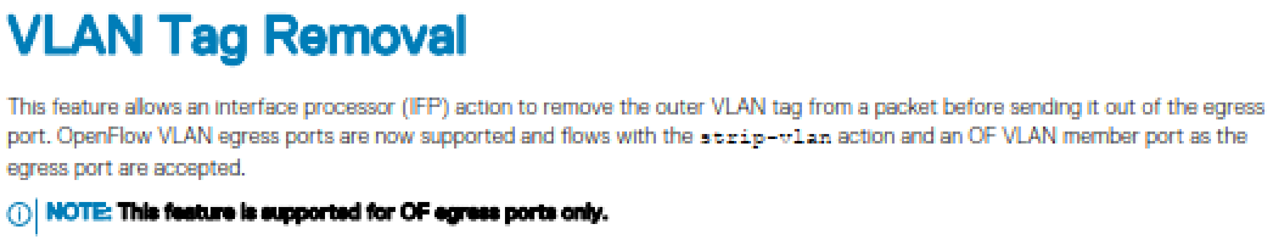
| Action | Associated Data | Description |
|---|---|---|
| Push VLAN header | Ethertype | Push a new VLAN header onto the packet.<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x8100 and 0x88a8 should be used. |
| Pop VLAN header | - | Pop the outer-most VLAN header from the packet. |
| Push MPLS header | Ethertype | Push a new MPLS shim header onto the packet.<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x8847 and 0x8848 should be used. |
| Pop MPLS header | Ethertype | Pop the outer-most MPLS tag or shim header from the packet.<br>The Ethertype is used as the Ethertype for the resulting packet (Ethertype for the MPLS payload). |
| Push PBB header | Ethertype | Push a new PBB service instance header (I-TAG TCI) onto the packet (see A.2.5).<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x88E7 should be used. |
| Pop PBB header | - | Pop the outer-most PBB service instance header (I-TAG TCI) from the packet (see A.2.5). |

Table˜6: Push/pop tag actions.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 23.

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td>[g] remove path or link information header when the incoming packet is forwarded to the external network, and</td><td>
The Exemplary Dell Switches have a forwarding unit configured to remove path or link information header when the incoming packet is forwarded to the external network.

*See, e.g.*,

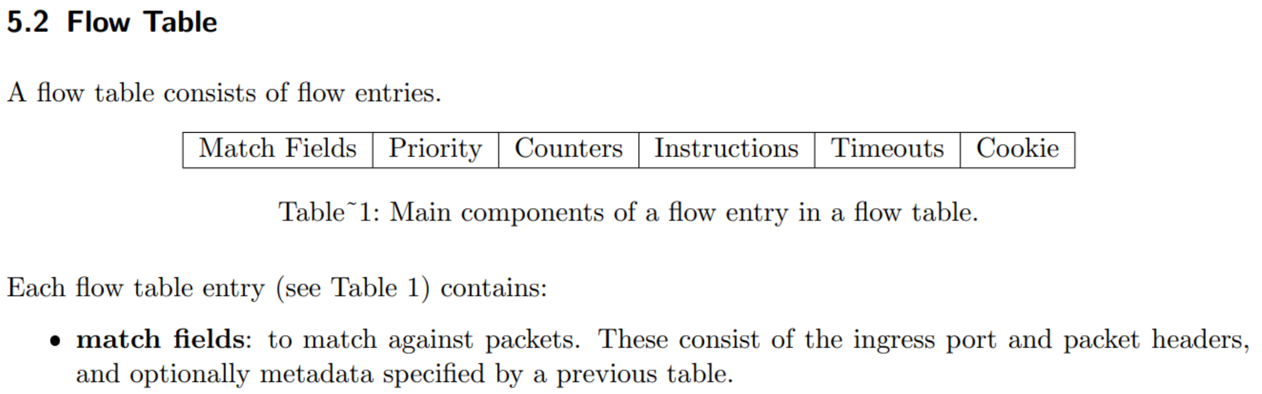


VLAN Tag Removal

This feature allows an interface processor (IFP) action to remove the outer VLAN tag from a packet before sending it out of the egress port. OpenFlow VLAN egress ports are now supported and flows with the `strip-vlan` action and an OF VLAN member port as the egress port are accepted.

ⓘ **NOTE: This feature is supported for OF egress ports only.**

Dell OpenFlow Deployment and User Guide 4.0, p. 25.



**5.2 Flow Table**

A flow table consists of flow entries.

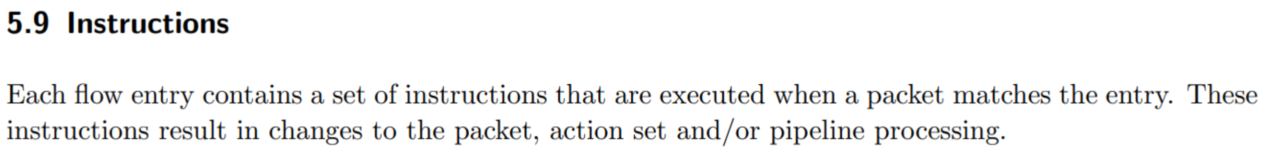
<table>
<tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr>
</table>

Table˜1: Main components of a flow entry in a flow table.

Each flow table entry (see Table 1) contains:

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.

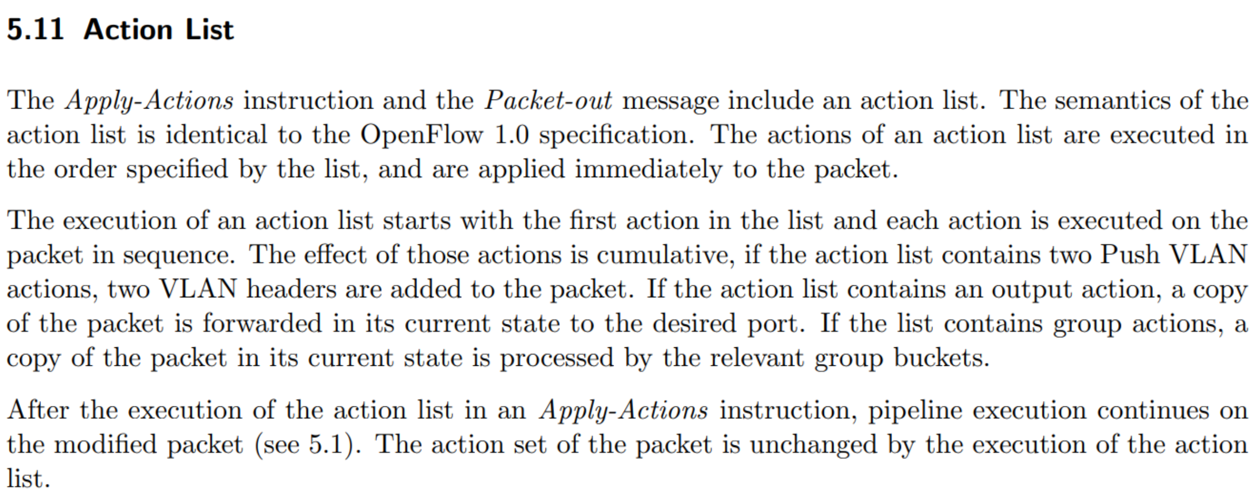


**5.9 Instructions**

Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.
</td></tr>
</table>

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td></td><td>

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 19.

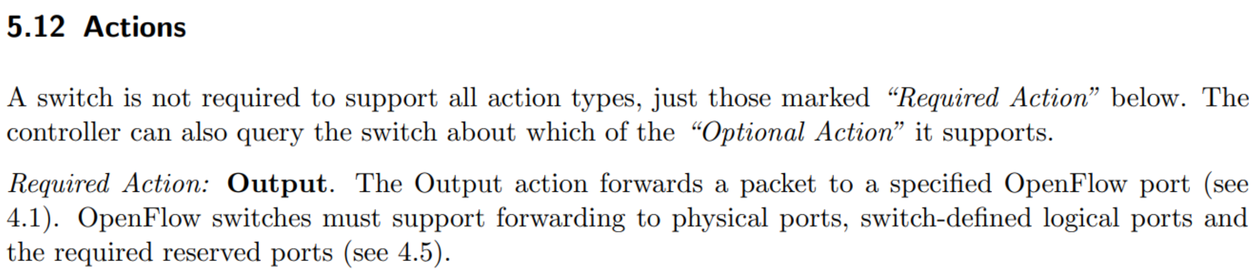


### 5.11 Action List

The *Apply-Actions* instruction and the *Packet-out* message include an action list. The semantics of the action list is identical to the OpenFlow 1.0 specification. The actions of an action list are executed in the order specified by the list, and are applied immediately to the packet.

The execution of an action list starts with the first action in the list and each action is executed on the packet in sequence. The effect of those actions is cumulative, if the action list contains two Push VLAN actions, two VLAN headers are added to the packet. If the action list contains an output action, a copy of the packet is forwarded in its current state to the desired port. If the list contains group actions, a copy of the packet in its current state is processed by the relevant group buckets.

After the execution of the action list in an *Apply-Actions* instruction, pipeline execution continues on the modified packet (see 5.1). The action set of the packet is unchanged by the execution of the action list.



### 5.12 Actions

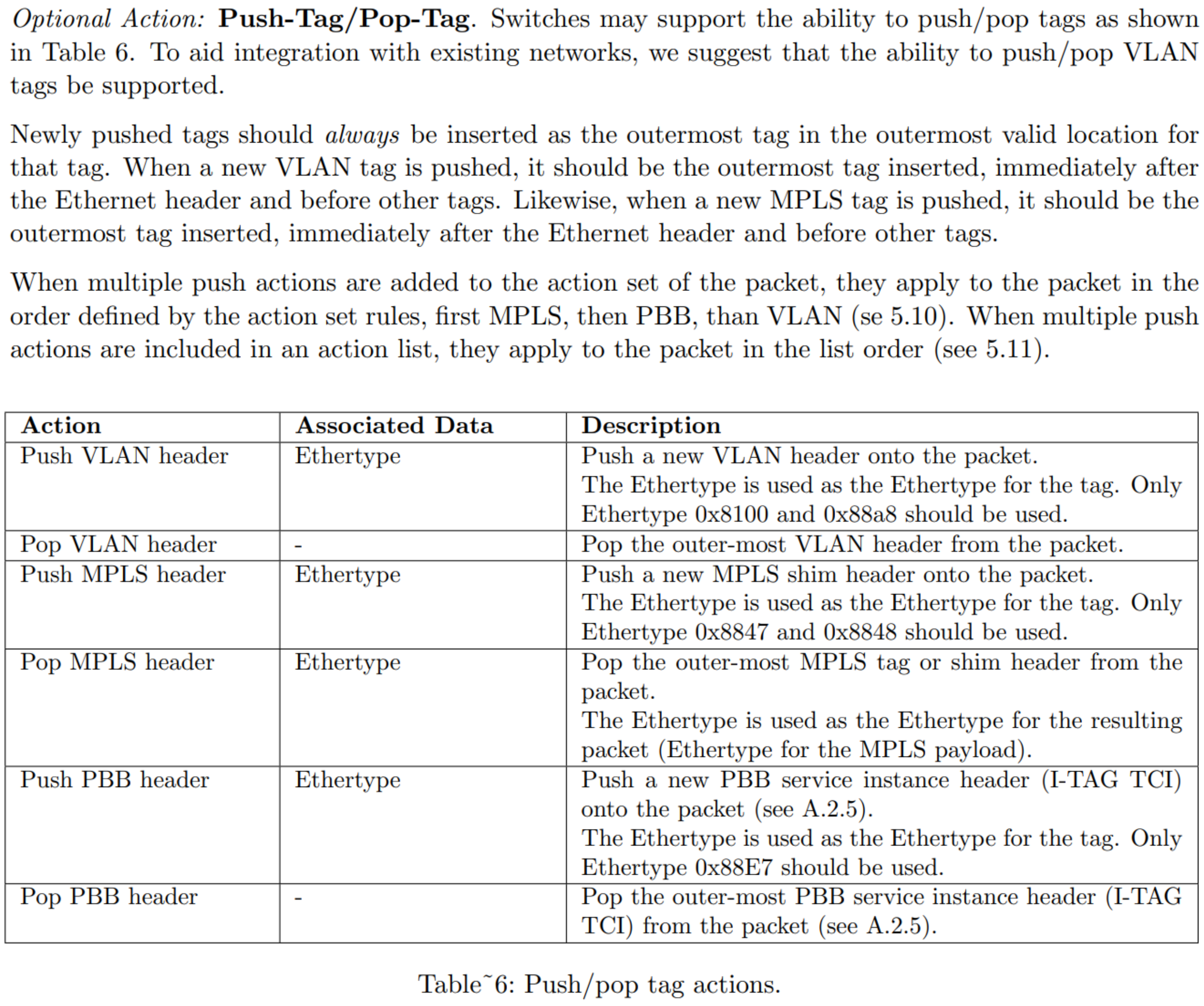
A switch is not required to support all action types, just those marked *"Required Action"* below. The controller can also query the switch about which of the *"Optional Action"* it supports.

*Required Action:* **Output**. The Output action forwards a packet to a specified OpenFlow port (see 4.1). OpenFlow switches must support forwarding to physical ports, switch-defined logical ports and the required reserved ports (see 4.5).

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 22.

</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|



*Optional Action:* **Push-Tag/Pop-Tag**. Switches may support the ability to push/pop tags as shown in Table 6. To aid integration with existing networks, we suggest that the ability to push/pop VLAN tags be supported.

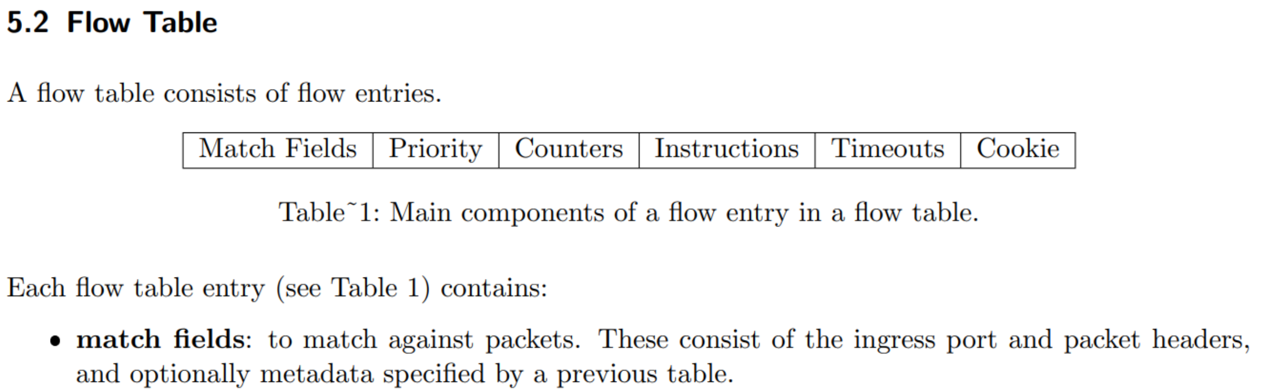
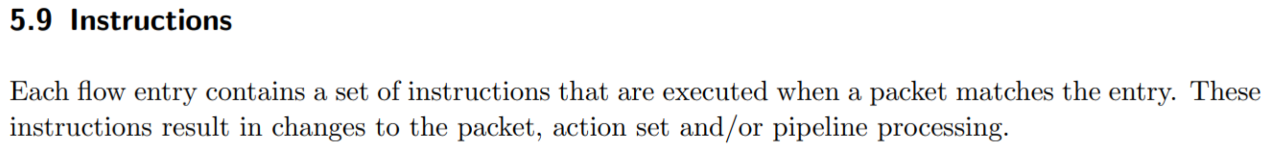
Newly pushed tags should *always* be inserted as the outermost tag in the outermost valid location for that tag. When a new VLAN tag is pushed, it should be the outermost tag inserted, immediately after the Ethernet header and before other tags. Likewise, when a new MPLS tag is pushed, it should be the outermost tag inserted, immediately after the Ethernet header and before other tags.

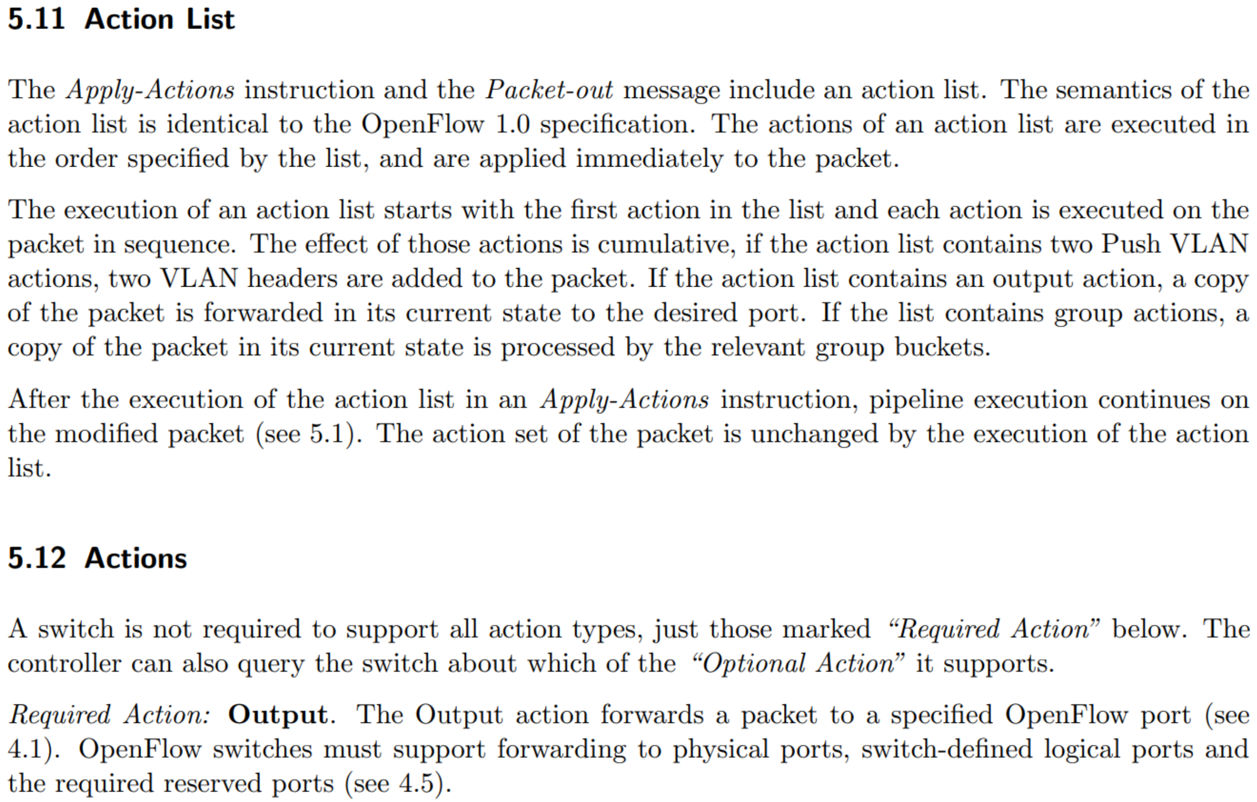
When multiple push actions are added to the action set of the packet, they apply to the packet in the order defined by the action set rules, first MPLS, then PBB, than VLAN (se 5.10). When multiple push actions are included in an action list, they apply to the packet in the list order (see 5.11).

| Action | Associated Data | Description |
|---|---|---|
| Push VLAN header | Ethertype | Push a new VLAN header onto the packet.<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x8100 and 0x88a8 should be used. |
| Pop VLAN header | - | Pop the outer-most VLAN header from the packet. |
| Push MPLS header | Ethertype | Push a new MPLS shim header onto the packet.<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x8847 and 0x8848 should be used. |
| Pop MPLS header | Ethertype | Pop the outer-most MPLS tag or shim header from the packet.<br>The Ethertype is used as the Ethertype for the resulting packet (Ethertype for the MPLS payload). |
| Push PBB header | Ethertype | Push a new PBB service instance header (I-TAG TCI) onto the packet (see A.2.5).<br>The Ethertype is used as the Ethertype for the tag. Only Ethertype 0x88E7 should be used. |
| Pop PBB header | - | Pop the outer-most PBB service instance header (I-TAG TCI) from the packet (see A.2.5). |

Table˜6: Push/pop tag actions.

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 23.

<table>
<tr><th>Claim 6</th><th>Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches</th></tr>
<tr><td>[h] to forward the incoming packet by using a path or link that corresponds to the identifier from among the plurality of identifiers.</td><td>The Exemplary Dell Switches have a forwarding unit configured to forward the incoming packet by using a path or link that corresponds to the identifier from among the plurality of identifiers.<br><br>*See, e.g.*,<br><br><br>**5.2 Flow Table**<br><br>A flow table consists of flow entries.<br><table><tr><td>Match Fields</td><td>Priority</td><td>Counters</td><td>Instructions</td><td>Timeouts</td><td>Cookie</td></tr></table><br>Table˜1: Main components of a flow entry in a flow table.<br><br>Each flow table entry (see Table 1) contains:<br>• **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 14.<br><br><br>**5.9 Instructions**<br><br>Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 19.</td></tr>
</table>

| Claim 6 | Exemplary Dell Switches: Dell S3048-ON, S3100, S4048-ON, S4048T-ON, S6100-ON, S6010-ON, S6000-ON, S4810, S4820T, S5000, S6000, Z9100, Z9500, FN IOM, and MXL switches |
|---|---|
|  | <br>**5.11 Action List**<br>The *Apply-Actions* instruction and the *Packet-out* message include an action list. The semantics of the action list is identical to the OpenFlow 1.0 specification. The actions of an action list are executed in the order specified by the list, and are applied immediately to the packet.<br>The execution of an action list starts with the first action in the list and each action is executed on the packet in sequence. The effect of those actions is cumulative, if the action list contains two Push VLAN actions, two VLAN headers are added to the packet. If the action list contains an output action, a copy of the packet is forwarded in its current state to the desired port. If the list contains group actions, a copy of the packet in its current state is processed by the relevant group buckets.<br>After the execution of the action list in an *Apply-Actions* instruction, pipeline execution continues on the modified packet (see 5.1). The action set of the packet is unchanged by the execution of the action list.<br>**5.12 Actions**<br>A switch is not required to support all action types, just those marked *"Required Action"* below. The controller can also query the switch about which of the *"Optional Action"* it supports.<br>*Required Action:* **Output**. The Output action forwards a packet to a specified OpenFlow port (see 4.1). OpenFlow switches must support forwarding to physical ports, switch-defined logical ports and the required reserved ports (see 4.5).<br>https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf, p. 22. |